UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/736,437	12/15/2003	David B. Kirk	NVDA/P000814	3433

26291 7590 04/05/2006

PATTERSON & SHERIDAN L.L.P.
595 SHREWSBURY AVE, STE 100
FIRST FLOOR
SHREWSBURY, NJ 07702

EXAMINER	
HSU, JONI	

ART UNIT	PAPER NUMBER
2628	

DATE MAILED: 04/05/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/736,437	KIRK ET AL.	
	Examiner	Art Unit	
	Joni Hsu	2628	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☐ Responsive to communication(s) filed on _____.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,2,11-16 and 23-35* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☒ Claim(s) *30-32* is/are allowed.

6)☒ Claim(s) *1,2,11-16,23-29 and 33-35* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

Response to Amendment

1. Applicant's arguments with respect to claims 1, 2, 11-16, 23-29, 34, and 35 have been considered but are moot in view of the new ground(s) of rejection.

2. Applicant's arguments, see pages 8-9, filed January 5, 2006, with respect to the rejection(s) of claim(s) 1, 2, 11-16, and 23-29 under 35 U.S.C. 103(a) have been fully considered and are persuasive. Therefore, the rejection has been withdrawn. However, upon further consideration, a new ground(s) of rejection is made in view of Mendelson (US005996060A).

3. With regard to Claims 1 and 11, Applicant argues that Bugnion (US006704925B1) teaches that when the cache is not flushed, there is a jump to a new translation, thereby guaranteeing that the stalled translation would never again be executed. Therefore, Bugnion cannot be combined with the Duluk references (US006288730B1, US006771264B1) to teach reading the x, y position to process a second fragment which depends on the first fragment process being completed without an intervening instruction to flush the graphics pipeline (page 8).

In reply, the Examiner agrees. However, new grounds of rejection are made in view of Mendelson.

4. With regard to Claim 23, Applicant argues that in the claims, the instruction is executed upon report that the execution of a previous fragment has been completed rather than waiting for the stalled or halted or locked instruction to execute from the queue, as taught in Duluk (US006288730B1) (pages 8-9).

In reply, the Examiner agrees. However, new grounds of rejection are made in view of Mendelson.

5. With regard to Claim 34, Applicant argues that the teaching that writes to the buffer can be to coordinate positions which are directly related to coordinate positions in the display is not disclosed in any of the references cited (page 9).

In reply, the Examiner agrees. However, new grounds of rejection are made in view of Ashburn (US006559852B1).

6. With regard to Claims 24 and 35, Applicant argues that neither the Duluk references nor Bugnion teach that other fragments are being processed while the second fragment is being delayed without the unlocking of the second fragment, thereby further speeding up the processing. Knittel (US006266733B1) also fails to teach anything about locking the processing of fragments in a graphics processor, and then unlocking the locked fragment upon completion of the processing of the first fragment (page 9).

In reply, the Examiner agrees. However, new grounds of rejection are made in view of Mendelson to teach locking the processing of instructions in a processor, and then unlocking the locked instruction upon completion of the processing of the first instruction (*the first instruction*

stores the result in register R1, Col. 4, lines 23-24; *the input operand R1 of the second instruction can be set only after the termination of the execution of the first instruction*, Col. 4, lines 27-35). Since Mendelson teaches this, Knittel can be combined to teach processing one or more additional fragments following processing the first fragment without unlocking the second fragment, as discussed below.

7. Applicant's arguments, see pages 6-9, filed January 5, 2006, with respect to Claims 30-33 have been fully considered and are persuasive. The 35 U.S.C. 103(a) rejections of Claims 30-33 has been withdrawn.

Claim Objections

8. Claim 23 is objected to because of the following informalities: Claim 23 recites the limitation "processing the second fragment in the fragment processing unit without flushing the pipeline between processing the first and second *segments*." Applicant is assumed to have meant "fragments" instead of "segments." Appropriate correction is required.

9. Claim 25 is objected to because of the following informalities: Claim 25 recites the limitation "specifying the position of the first *segment*." Applicant is assumed to have meant "fragment" instead of "segment." Appropriate correction is required.

Claim Rejections - 35 USC § 101

10. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

11. Claims 1, 2, 11-16, 27, and 33-35 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

12. Claim 1 recites an application programming interface comprising one or more program instructions. An application programming interface is a set of routines for building software applications, so an application programming interface is similar to a program. Claim 11 recites a fragment program comprising a sequence of instructions. Therefore, in both Claims 1 and 11, Applicant has merely recited a program; hence the claims are unpatentable. According to MPEP 2106 [R-3], IV, B, 1(a), computer programs claimed as computer listings per se, i.e., the descriptions of expressions of the programs, are not statutory processes. Both Claims 1 and 11 merely recite the program instructions and do not teach that the program instructions are executed to cause a functional change in the graphics pipeline; thus, the program is descriptive material per se and is not statutory. See MPEP 2106 [R-3], IV, B, 1(a) (computer programs claimed as computer listings per se, i.e., the descriptions of expressions of the programs, are not statutory processes). Given the absence of any practical effect or significant independent physical acts, Applicant's claims 1 and 11 fail to adequately define the claimed invention within the domain of patentable subject matter. Since Claims 2, 27 and 34 depend from Claim 1 and

Claims 12-16, 33, and 35 depend from Claim 11, these Claims are also rejected under 35 U.S.C. 101.

13. The Federal Circuit has in recent decisions applied the practical application test in determining whether the claimed subject matter is statutory under 35 U.S.C. 101. The practical application test requires that a "useful, concrete, and tangible result" be accomplished. See AT&T Corp. v. Excel Communications, Inc., 172 F.3d 1352, 1359-60, 50 USPQ2d 1447, 1452-53 (Fed. Cir. 1999); State Street Bank & Trust Co. v. Signature Financial Group Inc., 149 F.3d 1368, 1373, 47 USPQ2d 1596, 1600 (Fed. Cir. 1998). An "abstract idea" when practically applied is eligible for a patent. The Court in State Street, 149 F.3d at 1374, 47 USPQ2d at 1601 noted that "a process, machine, manufacture, or composition of matter employing a law of nature, natural phenomenon, or abstract idea is patentable subject matter even though a law of nature, natural phenomenon, or abstract idea would not, by itself, be entitled to such protection."

14. Since a computer program is merely a set of instructions capable of being executed by a computer, the computer program itself is not a process and is nonstatutory functional descriptive material. In contrast, a computer program that is claimed in a process where the computer is executing the computer program's instructions, which permit the program's functionality to be realized, and is thus statutory. See MPEP 2106 [R-3], IV, B, 1(a). The claimed invention as a whole must accomplish a practical application. That is, it must produce a "useful, concrete and tangible result." State Street, 149 F.3d at 1373, 47 USPQ2d at 1601-02.

Claim Rejections - 35 USC § 103

15. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

16. The factual inquiries set forth in *Graham* **v.** *John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

17. Claims 1, 2, and 27 are rejected under 35 U.S.C. 103(a) as being unpatentable over Duluk 1 (US006288730B1) in view of Duluk 2 (US006771264B1) in view of Mendelson (US005996060A).

18. With regard to Claim 1, Duluk 1 describes an application programming interface (Col. 5, lines 29-31) for a programmable graphics processor (Col. 5, lines 7-17; Col. 11, lines 64-65), comprising one or more program instructions (reorder logic, 2623-0, Figure 13b) within the programmable graphics processor (Col. 7, lines 40-41) to detect a position conflict for a position (Col. 14, lines 12-40). Duluk 1 describes that if a conflict is determined, the conflicting address

request is sent to the conflict queue (2604, Figure 13b; Col. 14, lines 24-26). Since the address request is in the conflict queue, the processor inherently waits until the address request that it is in conflict with has finished processing. Therefore, Duluk 1 discloses preventing a subsequent access of the position until the position conflict is resolved (Col. 14, lines 12-40). Duluk 1 describes determining if a memory conflict is likely to occur based upon the addresses contained in first level reorder queue (2603; Col. 14, lines 18-21), and therefore inherently includes instructions to read the x, y position to process a second fragment (*texture maps are stored in memory using texture tile addresses, texels are used to generate fragments*, Col. 9, lines 30-45). Duluk 1 describes that the address is added to first level reorder queue (2603; Col. 14, lines 21-24), and therefore inherently includes instructions to write the x, y position upon completion of a first fragment process (Col. 9, lines 30-45). Since there is a conflict if a second fragment attempts to access the same position that the first fragment is written before completion of the first fragment process (Col. 14, lines 12-40; Col. 9, lines 30-45), the processing of the second fragment depends of the first fragment process being completed.

However, the program instructions described by Duluk 1 are for reordering the memory addresses to be accessed (Col. 14, lines 1-19), and Duluk 1 does not explicitly teach that these program instructions are for configuring a fragment processor. However, a related patent, Duluk 2, describes that OpenGL defines a set of per-fragment operations (Col. 6, lines 34-37), and these per-fragment operations determine the manner in which the pixels are stored (Col. 6, lines 42-65). Therefore, Duluk 2 describes one or more program instructions to configure a fragment processor, and these per-fragment operations determine the manner in which the pixels are stored.

It would have been obvious to one of ordinary skill in this art at the time of invention by applicant to modify the device of Duluk 1 so that the program instructions are for configuring a fragment processor as suggested by Duluk 2 because Duluk 2 suggests that this is needed in order to determine the manner in which the pixels are stored (Col. 6, lines 34-37, 42-65), and also to perform various tests on the fragments, such as the ownership test, scissor test, alpha test, color test, stencil test, depth buffer test, to determine what operation needs to be performed on the fragment to modify the pixel in the frame buffer at that location (Col. 7, lines 8-22).

However, Duluk 1 and Duluk 2 do not teach that the instructions further includes one or more instructions to read the x, y position to process a second fragment which depends on the first fragment process being completed without an intervening instruction to flush the graphics pipeline. However, Mendelson describes writing the x, y position upon completion of a first instruction (*the first instruction stores the result in register R1*, Col. 4, lines 23-24) and reading the x, y position to process a second instruction which depends on the first instruction being completed (*the input operand R1 of the second instruction (the read instruction), can be set only after the termination of the execution of the first instruction (the write instruction)*, Col. 4, lines 27-35) without an intervening instruction to flush the pipeline (*as is well known to those versed in the art, "forwarding" is a technique that is normally used in the case that the output of a given instruction is used as an input of a succeeding instruction (i.e. read-after-write conflict), the forwarding technique affords to skip conventional "house cleaning" stages, that are normally executed upon termination of the execution of an instruction, by "injecting" the result to the waiting operand of the succeeding instruction (the read instruction), immediately upon the production thereof at the output of the execution unit*, Col. 6, lines 12-22).

It would have been obvious to one of ordinary skill in this art at the time of invention by applicant to modify the devices of Duluk 1 and Duluk 2 so that the instructions further includes one or more instructions to read the x, y position to process a second fragment which depends on the first fragment process being completed without an intervening instruction to flush the graphics pipeline as suggested by Mendelson because Mendelson suggests that if a second instruction depends on the completion of the first instruction, the reading of the x, y position to process the second instruction can commence only upon the completion of the first instruction in order to avoid erroneous operation (Col. 4, lines 27-52), and there is no intervening instruction to flush the pipeline since the second instruction depends on the first instruction, so the first instruction cannot be flushed (Col. 6, lines 10-22).

19. With regard to Claim 2, Duluk 1 describes that a program instruction (2623-0, Figure 13b) receives memory addresses, and for each memory address received, conflict detection block (2602) determines if a memory conflict is likely to occur based upon the addresses contained in first level reorder queue (2603) (Col. 14, lines 12-40). In order to determine if a memory conflict is likely to occur, the addresses received by a program instruction must inherently include a source location and a destination location.

20. With regard to Claim 27, Duluk 1 describes that the position comprises a region including a plurality of pixels *(texture tile addresses*, Col. 9, lines 30-45; *tiles have 16x16 pixels*, Col. 5, lines 53-56).

21. Claims 11, 12, 15, 16, 23, 25, 26, 28, and 29 are rejected under 35 U.S.C. 103(a) as being unpatentable over Duluk 1 (US006288730B1) in view of Mendelson (US005996060A).

22. With regard to Claim 11, Duluk 1 describes a fragment program for processing fragment data in a fragment processing pipeline (Col. 5, lines 14-17, 42-67), comprising a sequence of instructions comprising shading a first fragment (*texture values are sent downstream to a shading block which may then combine the texture value with other image information such as lighting to generate the final color value for a fragment*, Col. 8, lines 14-18) associated with a destination location in a buffer; storing the result of the first fragment shading in the destination location in the buffer (Col. 11, line 64-Col. 12, line 7). Duluk 1 describes that if a conflict for shading a second fragment is determined, the conflicting address request is sent to the conflict queue (2604, Figure 13b; Col. 14, lines 24-26). Since the address request for shading the second fragment is in the conflict queue, the processing of the second fragment is not yet executed and is thus interrupted and the processing of the second fragment is later resumed, processing of the second fragment including reading the destination location in the buffer. Since there is a conflict if a second fragment attempts to access the same position that the first fragment is written before completion of the first fragment process (Col. 14, lines 12-40; Col. 9, lines 30-45), the processing of the second fragment depends of the first fragment process being completed.

However, Duluk 1 does not teach that shading of the second fragment is based in part of reading the result of the first fragment shading from the destination location, the conflict is a read after write position conflict, and that on completing processing of the first instruction and storing the result in the destination location in the buffer resuming processing of the second fragment

without an intervening instruction to flush the fragment processing pipeline. However, Mendelson describes that the processing of the second instruction is based in part of reading the result of the first instruction from the destination location, the conflict is a read after write position conflict and on completing processing of the first instruction and storing the result in the destination location in the buffer (Col. 4, lines 23-24) resuming processing of the second instruction including reading the destination location in the buffer (Col. 4, lines 27-35) without an intervening instruction to flush the pipeline (Col. 6, lines 12-22). This would be obvious for the same reasons given in the rejection for Claim 1.

23. With regard to Claim 12, Duluk 1 describes that the destination location includes a buffer identifier corresponding to one of several buffers (Col. 12, lines 24-41).

24. With regard to Claim 15, Duluk 1 describes fragment program instructions to configure the fragment processing pipeline to perform raster operations (Col. 2, lines 21-25; Col. 6, lines 42-67).

25. With regard to Claim 16, Duluk 1 describes that the raster operations are performed using fragment data represented in a floating-point data format (Col. 2, lines 15-18, 21-25; Col. 8, lines 19-41).

26. With regard to Claim 23, Duluk 1 describes a method for processing fragments in a graphics processor pipeline comprising providing a fragment processing unit within the graphics

processor pipeline; receiving a first fragment associated with a position by the fragment processing unit; processing the first fragment associated with the position to obtain a processed first fragment (*texture pipeline, texture unit 1200 receives texture coordinates for individual fragments, generates a texture value for each fragment*, Col. 8, lines 9-14); receiving a second fragment associated with the position by the fragment processing unit; writing the processed first fragment to a graphics memory; and processing the second fragment in the fragment processing unit (Col. 14, lines 24-27; Col. 7, lines 40-43). Since there is a conflict if a second fragment attempts to access the same position that the first fragment is written before completion of the first fragment process (Col. 14, lines 12-40; Col. 9, lines 30-45), the processing of the second fragment depends of the first fragment process being completed.

However, Duluk 1 does not teach interlocking the second fragment in part subject to completion of the processing of the first fragment; unlocking the second fragment after writing the processed first fragment; and processing the second fragment without flushing the pipeline between processing the first and second fragments. However, Mendelson describes interlocking the second instruction in part subject to completion of the processing of the first instruction; unlocking the second instruction after writing the processed first instruction to a memory (Col. 4, lines 23-35); and processing the second instruction without flushing the pipeline between processing the first and second instructions (Col. 6, lines 12-22). This would be obvious for the same reasons given in the rejection for Claim 1.

27. With regard to Claim 25, Duluk 1 describes specifying the position of the first fragment as source data for subsequent processing of fragments (*address is added to first level reorder*

queue 2603, to allow for conflict checking of subsequently received addresses, Col. 14, lines 18-24).

28. With regard to Claim 26, Duluk 1 does not teach that the interlocking step comprises reaching the source data prior to processing the second fragment to prevent writing to the position. However, Mendelson describes that the interlocking step comprises reaching the source data prior to processing the second instruction to prevent writing to the position (Col. 4, lines 23-35). This would be obvious for the same reasons given in the rejection for Claim 1.

29. With regard to Claim 28, Duluk 1 does not teach checking a location in graphics memory for the processed first fragment prior to unlocking the second fragment. However, Mendelson describes checking a location in memory for the processed first instruction prior to unlocking the second instruction (Col. 4, lines 23-35). This would be obvious for the same reasons given in the rejection for Claim 1.

30. With regard to Claim 29, Duluk 1 describes processing the first and either the second or the additional fragments in parallel (Col. 9, lines 20-29).

31. Claim 13 is rejected under 35 U.S.C. 103(a) as being unpatentable over Duluk 1 (US006288730B1) and Mendelson (US005996060A) in view of Wood (US006204856B1).

Duluk 1 and Mendelson are relied upon for the teachings as discussed above relative to Claim 11.

However, Duluk 1 and Mendelson do not teach fragment program instructions to configure the fragment processing pipeline to perform depth buffering prior to shading. However, Wood describes fragment program instructions to configure the fragment processing pipeline to perform depth buffering prior to shading (Col. 1, lines 22-24; Col. 9, lines 64-67; Col. 12, lines 1-6).

It would have been obvious to one of ordinary skill in this art at the time of invention by applicant to modify the devices of Duluk 1 and Mendelson to include fragment program instructions to configure the fragment processing pipeline to perform depth buffering prior to shading as suggested by Wood because Wood suggests the advantage of reducing the number of attributes to be calculated (Col. 1, lines 62-64; Col. 11, line 62-Col. 12, line 6).

32. Claim 14 is rejected under 35 U.S.C. 103(a) as being unpatentable over Duluk 1 (US006288730B1) and Mendelson (US005996060A) in view of Isard (US 20040207623A1).

Duluk 1 and Mendelson are relied upon for the teachings as discussed above relative to Claim 11.

However, Duluk 1 and Mendelson do not teach fragment program instructions to configure the fragment processing pipeline to perform depth peeling. However, Isard describes fragment program instructions to configure the fragment processing pipeline to perform depth peeling [0017, 0055].

It would have been obvious to one of ordinary skill in this art at the time of invention by applicant to modify the devices of Duluk 1 and Mendelson to include fragment program instructions to configure the fragment processing pipeline to perform depth peeling as suggested

by Isard because Isard suggests that depth peeling is needed to render shadows cast by transparent objects [0055].

33. Claims 24 and 35 are rejected under 35 U.S.C. 103(a) as being unpatentable over Duluk 1 (US006288730B1) and Mendelson (US005996060A) in view of Knittel (US006266733B1).

34. With regard to Claim 24, Duluk 1 and Mendelson are relied upon for the teachings as discussed above relative to Claim 23.

However, Duluk 1 and Mendelson do not specifically teach processing one or more additional fragments following processing the first fragment without unlocking the second fragment. However, Knittel describes that cases occasionally arise where the fetching of a mini-block from one bank of one DRAM at the end of one row is followed in quick succession by the fetching of a mini-block from the same bank of the same DRAM module at the start of the next row. To avoid this problem, there is a system for reversing the direction of the read-out of a row of mini-blocks when the next mini-block to be read out would result in a fetch from the same or a conflicting bank of the DRAM memory (Col. 3, line 60-Col. 4, line 2). Since the system is reversing the direction of the read-out, it is inherently preventing the reading of the conflicting block and reading another block on the other end of the row. Therefore, Knittel discloses processing one or more additional voxel data following processing the first block of voxel data without unlocking the second block of voxel data (Col. 3, line 60-Col. 4, line 2; Col. 3, lines 49-50).

It would have been obvious to one of ordinary skill in this art at the time of invention by applicant to modify the devices of Duluk 1 and Mendelson to include processing one or more additional fragments following processing the first fragment without unlocking the second fragment as suggested by Knittel because Knittel suggests the advantage of avoiding conflicts while avoiding the delay cycles of the prior art (Col. 3, lines 9-25).

35. With regard to Claim 35, Claim 35 is similar in scope to Claim 24, and therefore is rejected under the same rationale.

36. Claim 34 is rejected under 35 U.S.C. 103(a) as being unpatentable over Duluk 1 (US006288730B1), Duluk 2 (US006771264B1), and Mendelson (US005996060A) in view of Ashburn (US006559852B1).

Duluk 1, Duluk 2, and Mendelson are relied upon for the teachings as discussed above relative to Claim 1. Duluk 1 describes detecting a position conflict for an x, y position (*determines if a memory conflict is likely to occur based upon the addresses*, Col. 14, lines 18-21), wherein the x, y position comprises coordinates within a texture map (Col. 9, lines 30-45).

However, Duluk 1, Duluk 2, and Mendelson do not teach that the x, y position comprises coordinates within a display. However, Ashburn describes detecting a position conflict for an x, y position (Col. 4, lines 5-9; *accesses to the same pixel location had to be placed in separate batches; otherwise the result would be a "pixel collision"*, Col. 3, lines 54-57), wherein the x, y position comprises coordinates within a display (Col. 3, lines 11-13).

It would have been obvious to one of ordinary skill in the art at the time of invention by applicant to modify the devices of Duluk 1, Duluk 2, and Mendelson so that the x, y position comprises coordinates within a display as suggested by Ashburn. Ashburn suggests that over time, the resolution capabilities of display devices have increased, and consequently so has the amount of information that must be stored in the frame buffer memory. In addition, refresh cycles of display devices have become shorter. The result has been that access rates for modern frame buffer memories have become extremely high (Col. 1, lines 52-56). Modern applications utilize greater depth complexity, and thus pixel collision occur more frequently than in the past (Col. 3, lines 62-65), and therefore there is a need for detecting position conflicts within a display.

Allowable Subject Matter

37. Claims 30-32 are allowed.

38. Claim 33 objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

The following is a statement of reasons for the indication of allowable subject matter:

39. The prior art taken singly or in combination do not teach or suggest a fragment processing pipeline configured to handle read-after-write hazards during execution of shader programs including an instruction to write a location in graphics memory, an instruction to check

the location in a conflict detection unit and a subsequent instruction to read a location in graphics memory without an intervening instruction to flush the fragment processing pipeline based on the check of the conflict detection unit, as recited in Claim 30. Claims 31 and 32 depend from Claim 30, and therefore also contain allowable subject matter.

The prior art also does not teach outputting write position information to the conflict detection unit confirming that shading of the first fragment and writing to the buffer is complete, as recited in Claim 33.

40. The closest prior art (Duluk 1 US006288730B1) teaches a programmable graphics processor for execution of program instructions (Col. 1, lines 64-67), comprising a read interface (1210, Figure 2) configured to read data from a graphics memory (1213; Col. 8, lines 59-60); a fragment processing unit configured to receive fragments (Col. 7, lines 40-41), each fragment associated with a position (Col. 8, lines 23-24), and the data from the graphics memory and generate processed fragments (Col. 7, lines 40-43); a conflict detection unit (2602, Figure 13b) configured to selectively store the position associated with each fragment and generate a position conflict status (Col. 14, lines 18-24); a write interface (1210, Figure 2) configured to write the processed fragments to the graphics memory (Col. 8, lines 59-60); and a fragment processing pipeline (Col. 5, lines 14-17; Col. 8, lines 19-27) configured to handle conflicts during execution of shader programs (Col. 6, lines 61-65) including an instruction to write a location in graphics memory, an instruction to check the location in a conflict detection unit and a subsequent instruction to read a location in graphics memory (Col. 14, lines 18-24). However, Duluk 1 does not teach that the conflicts are read-after-write hazards, the read is done without an interleaving

instruction to flush the fragment processing pipeline based on the check of the conflict detection unit, and outputting write position information to the conflict detection unit confirming that shading of the first fragment and writing to the buffer is complete.

41. Another prior art (Mendelson US005996060A) teaches a processing pipeline configured to handle read-after-write hazards including writing a location in memory (Col. 4, lines 23-24), and a subsequent instruction to read a location in memory (Col. 4, lines 27-35) without an interleaving instruction to flush the processing pipeline (Col. 6, lines 12-22). However, Mendelson does not teach that the processing pipeline processes fragments, the read-after-write hazards are handled during execution of shader programs, and outputting write position information to the conflict detection unit confirming that shading of the first fragment and writing to the buffer is complete.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Joni Hsu whose telephone number is 571-272-7785. The examiner can normally be reached on M-F 8am-5pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Ulka Chauhan can be reached on 571-272-7782. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

JH

ULKA CHAUHAN
SUPERVISORY PATENT EXAMINER